Module 1:   
Assignment Solution

Q1. List out the key difference between a centralized version control system and distributed version control system  
A1. Centralized version control system has the idea that there is a single centralized copy and the programmers will commit and pull from that centralized server only. Other programmers can see the changes committed. Every programmer will pull, push on this copy only and needs to be connected to this central copy.  
  
Distributed systems doesn’t rely on central servers to store all the version. Here every developer clone’s a copy of the project and hence all the version are now available on the local hard disk. Keeping a centralized repository is optional. Pulling and pushing are methods to get the code and push the code to the repository respectively and hence making it available for other developer to get the code. The programmer can work on his local clone repository and commit changes to this local repository without being connected to internet. When a sensible amount of changes have been done the developer can push the code to the repository for it to be available to other developers. This step requires internet connection.   
  
Branches can also be created in DVCS and developer can develop some feature on that branch without making it available for other users and merge this branch with main branch when he fell the feature is developed.  
  
  
Q2. List down any two centralized version control system and 2 distributed version control system.  
A2. CVCS: SVN, IBM ClearCase  
 DVCS: Git, Mecurial.  
  
Q3. What are the advantages of git VCS over other VCS?  
A3. The major difference between Git and any other is the way Git thinks about its data. Conceptually, most other systems store information as a list of file-based changes. These systems think of the information they keep as a set of files and the changes made to each file over time. Git doesn’t think of or store its data this way. Instead, Git thinks of its data more like a set of snapshots of a miniature filesystem. Every time you commit, or save the state of your project in Git, it basically takes a picture of what all your files look like at that moment and stores a reference to that snapshot. To be efficient, if files have not changed, Git doesn’t store the file again, just a link to the previous identical file it has already stored. Git thinks about its data more like a **stream of snapshots.**  
  
Q4. What are the different states of a file in the Git VCS?  
A4. There are two high level staged :  
 Tracked FIles  
 Untracked Files  
 Tracked file can have 3 stages:  
 Unmodified files  
 Modified Files  
 Staged FIles